

거대 언어모델의 시대에서 살아가는 개발자

권민철

2023.10.27

Abstract

프로그래머의 역할은 인공지능으로 완전히 대체될 수 있는지에 대한 논의가 있다. 새로운 코드 작성에서는 인공지능의 자연어 이해와 코드 생성 능력에도 오류가 발생하며, 기존 코드 유지 보수에서는 코드의 의도와 비즈니스 규칙을 이해하고 적절한 수정을 가하는 것이 어렵다. 따라서, 인공지능은 개발자의 역할을 보조하는 도구로 활용되어야 하며, 개발자의 맥락 이해와 검증 작업은 여전히 필요하다.

‘프로그래머(Programmer)라는 직업은 5년 안에 사라질 것이다.’ 스테빌리티 에이아이(Stability AI)의 대표인 이마드 모스타크가 한 말이다. 챗지피티(ChatGPT)의 등장으로 급격히 성장한 거대 언어모델(Large Language Model)은 다양한 분야로 진출하였고, 그 중 하나인 깃허브(Github)에서 출시한 코파일럿(Copilot)은 기존 프로그래밍 환경을 뒤바꿔 놓는, 그야말로 혁신이었다. 코드를 작성하다가 막히는 부분이 있으면 잠시 기다리고, 자동으로 작성된 코드를 사용하면 되었다. 심지어 아이폰 기본 알람 앱(Application)의 사진을 보여주면, 바로 사용할 수 있는 코드를 제공해주는 수준까지 발전하였다. 과연 기존 개발자의 자리를, 이러한 인공지능이 대체할 수 있을까? 개발자는 크게 새로운 코드를 작성하고, 기존 코드를 유지 보수하는 역할을 한다고 볼 수 있다. 각 측면에서 개발자의 대체 가능성에 대해 살펴보자.

먼저, 새로운 코드 작성 면에서 살펴보자. 코드가 어떤 작업을 해야 할지는 자연어로 표현되어 있고, 이를 이해하고 실제 코드로 기술하는 능력이 필요하다. 인공지능의 경우, 원하는 작업을 기술한 자연어 자체가 부정확하기 때문에 이를 이해하는 과정과 실제 코드로 옮기는 과정에서 다양한 오류를 범한다. 의도와 다른 코드가 생성되거나, 문법 오류 혹은 보안 취약점을 포함하는 코드를 작성하는 등의 문제가 발생할 수 있다. 예를 들어, 거대 언어모델을 허용한 수업의 과제를 위해 코파일럿을 활용하고 있는데, 생성된 코드를 살펴보면 컴파일(compile)조차 되지 않거나, 테스트 케이스(test case) 일부를 통과하지 못하는 경우가 잦았다. 이를 적절히 수정하기 위해서는 결국 과제에 대응되는 개념을 스스로 공부하여 잘못된 부분을 고쳐야 했다. 반복되는 코드나 간단한 정렬 알고리즘 등의 검색에 낭비되는 시간을 줄여주는 역할을 할 수 있을지는 몰라도, 생성된 코드를 검증하고 고쳐야 하는 것은 여전히 인간만이 할 수 있는 일이다.

다음으로, 기존 코드 유지 보수 면에서 살펴보자. 기존 코드를 이해하고 수정하는 역할은 개발자의 전문 분야이다. 인공지능은 주어진 코드를 분석하고 수정하는 능력은 갖추고 있을지라도, 코드의 의도나 비즈니스 규칙을 이해하고 적절한 수정을 가하는 것은 어려울 수 있다. 소프트웨어(Software)의 경우 유지 보수를 위해 지라(Jira)와 같은 이슈 트래커 (Issue Tracker)를 활용한다. 하나의 소프트웨어라 하더라도, 개발은 많은 사람의 협업으로 이루어지기 때문에, 다른 사람의 코드의 맥락을 이해하기 위해서는 코드 작성 외에도 협업 도구와 사람 간의 소통이 필요하다. 맥락을 이해하지 못하고 생성된 코드를 사용하게 된다면, 수업시간에 배웠던 2009년에 발견된 버퍼(Buffer)가 넘칠 수 있던 취약점이 그대로 2017년에 다시 발견되는 등, 맥락 없이 남의 코드를 복제하는 것과 다를 바가 없다. 먼 훗날 코드의 맥락까지 파악할 수 있는 생성 인공지능이 나올지도 모르는 일이지만, 인공지능이 현재의 개발자를 완전히 대체할 수 있는지는 불분명하다.

새로운 코드 작성과 기존 코드 유지보수, 두 가지 측면에서 개발자의 대체 가능성에 대해 살펴 보았을 때, 여전히 개발자의 역할은 필요하다. 거대 언어모델의 계속되는 학습과 생성된 코드를 검증하는 단계를 추가하는 등 성능을 개선하기 위한 작업은 꾸준히 진행되고 있지만, 결국 언어 모델이 생성한 것이라는 점에서 오류가 없다는 결과를 검증하는 것은 인간의 몫이다. 코파일럿과 같은 코드 작성 도구는 능률을 향상하기 위한 보조 수단으로 인식하고 활용하는 것이 인공지능의 시대에서 살아가고 있는 개발자가 취해야 할 적절한 입장이라 생각한다.